Whole-Body Control

Notes by: Jun Young Kim

* It is a personal notation and formulation for studying whole-body control; the note is based on various references, including papers and books. Therefore, the notation might be mixed section by section.

* **Vectors** are expressed in bold and lowercase, **matrices** are expressed in bold and uppercase, and scalars are lowercase; use \mathcal to represent the frame. Underline in green means the answer from ChatGPT. Undeline in yellow means exact copy from the reference papers.

Nomenclature

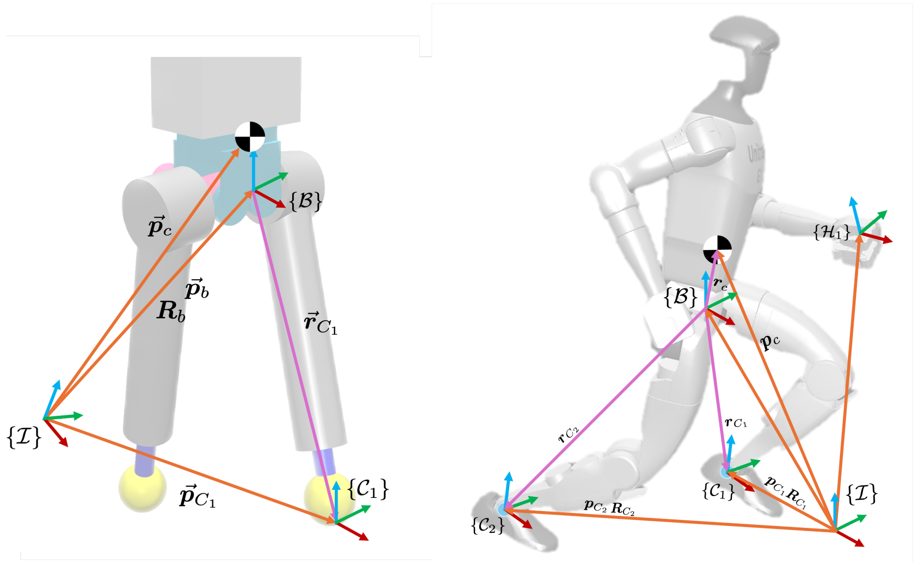

Figure 1: Nomenclature on pointfoot biped and humanoid robot, G1

- $\{\mathcal{I}\}$ is Inertial (World) frame, $\{\mathcal{B}\}$ is Body (Base) frame, $\{\mathcal{H}_k\}$ is k-th hand frame, $\{\mathcal{C}_k\}$ is k-th contact frame
- ${}^k\boldsymbol{x}_{mn}$ is a vector from $\{\mathcal{M}\} \rightarrow \{\mathcal{N}\}$ expressed in $\{\mathcal{K}\}$
- $\boldsymbol{p}_{jk}$ is position vector $O_J \rightarrow O_K$ expressed in $\{\mathcal{I}\}$, if $\{J\}$ frame is equal to inertial frame, then $\boldsymbol{p}_{jk} = \boldsymbol{p}_k$
- $\boldsymbol{r}_{jk}$ is position vector $O_J \rightarrow O_K$ represented in $\{\mathcal{B}\}$, if $\{J\}$ frame is equal to inertial frame, then $\boldsymbol{r}_{jk} = \boldsymbol{r}_k$

Whole-body Control

To maintain the robot balance and track the desired motion tasks (trajectories) in dynamic environments, it is now common to use whole-body control (WBC). WBC is the method that enables the control of a robot's full body while adhering to various physical or contact constraints, operating in a closed-loop fashion. As listed in [1], depending on the robot's structure and the control methodology, there are various approaches that can be taken.

Normally, we utilize Quadratic Programming (QP) to find optimal joint acceleration and contact force that considers the dynamics of the robot and other physical constraints.

$$\boldsymbol{z} = \begin{bmatrix} \ddot{\boldsymbol{\xi}}^T & \boldsymbol{\lambda}^T & \boldsymbol{\tau}^T \end{bmatrix}^T \text{ or } \begin{bmatrix} \ddot{\boldsymbol{\xi}}^T & \boldsymbol{f}^T \end{bmatrix}^T \text{ or } \begin{bmatrix} \ddot{\boldsymbol{\xi}}^T & \boldsymbol{\beta}^T \end{bmatrix}^T \tag{1}$$

In general, we assume that each control objective can be expressed as a linear combination of the decision variables of our problem. The common approach to control the robot can be divided by using weighted WBC [2], hierarchical WBC [3], or task-prioritized WBC [4].

Optimal Control Problem

Dynamic Equations of Motion

Dynamic equations of motion of the floating-based model can be expressed as follows:

$$\underbrace{\begin{bmatrix} \boldsymbol{M}_b \\ \boldsymbol{M}_j \end{bmatrix}}_{\boldsymbol{M}} \ddot{\boldsymbol{\xi}} + \underbrace{\begin{bmatrix} \boldsymbol{C}_b \\ \boldsymbol{C}_j \end{bmatrix}}_{\boldsymbol{C}} \dot{\boldsymbol{\xi}} + \underbrace{\begin{pmatrix} \boldsymbol{g}_b \\ \boldsymbol{g}_j \end{pmatrix}}_{\boldsymbol{g}} = \underbrace{\begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{\tau}_j \end{pmatrix}}_{\boldsymbol{\tau}} + \sum_{k}^{N_c} \underbrace{\begin{bmatrix} \boldsymbol{X}_{bC_k}^T \\ \boldsymbol{J}_{bC_k}^T \end{bmatrix}}_{\boldsymbol{X}_{C_k}^T} \boldsymbol{\lambda}_{C_k} \tag{2}$$

where,

$$\dot{\boldsymbol{\xi}} = \begin{pmatrix} \boldsymbol{v}_b \\ \dot{\boldsymbol{q}}_j \end{pmatrix}, \boldsymbol{X}_{bC_k}^T = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0} \\ [\boldsymbol{r}_{C_k}]_\times & \mathbf{I}_3 \end{bmatrix}, \boldsymbol{\lambda}_{C_k} = \begin{pmatrix} \boldsymbol{f}_{C_k} \\ \boldsymbol{\mu}_{C_k} \end{pmatrix} = \bar{\boldsymbol{R}}_{C_k} \bar{\boldsymbol{W}}_k \boldsymbol{\beta}_k \tag{3}$$

where $\boldsymbol{M}$, $\boldsymbol{C} \in \mathbb{R}^{(n+6)\times(n+6)}$ are an inertia and a Coriolis and Centrifugal matrices of the body and $\boldsymbol{g} \in \mathbb{R}^{(n+6)}$ is a gravitational force vector of the body. $\boldsymbol{X}_{bC_k}$, $\boldsymbol{J}_{bC_k}$ are the part of Jacobian where the former is related to the body velocity and the latter is related to the joint velocity.

Solve control problem

Using the inverse dynamics equation as follows:

$$\boldsymbol{\tau}_j = \boldsymbol{M}_j \ddot{\boldsymbol{\xi}} + \boldsymbol{C}_j \dot{\boldsymbol{\xi}} + \boldsymbol{g}_j - \sum_{k}^{N_c} \boldsymbol{J}_{bC_k}^T \boldsymbol{\lambda}_{C_k} \tag{4}$$

we computes the command joint torque.

Tasks

Centroidal Dynamics Task

Newton-Euler Equations are a set of equations that describe the motion of a rigid body. They are derived from *Newton's laws of motion*:

$$m(\ddot{\boldsymbol{p}}_c + \boldsymbol{g}) = \sum_{i=1}^{n_c} \boldsymbol{f}_i \tag{5}$$

and *Euler's equations of motion*:

$$\begin{gathered} \frac{d}{dt}(\boldsymbol{I}\boldsymbol{\omega}) = \boldsymbol{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\boldsymbol{I}\boldsymbol{\omega}) \approx \boldsymbol{I}\dot{\boldsymbol{\omega}} \\ \boldsymbol{I}\dot{\boldsymbol{\omega}} = \sum_{i=1}^{n_c} \boldsymbol{r}_i \times \boldsymbol{f}_i \end{gathered} \tag{6}$$

where m is the mass of the body, $\boldsymbol{I}$ is the inertia of the body, $\boldsymbol{\omega}$ is the angular velocity of the body, and $\dot{\boldsymbol{\omega}}$ is the angular acceleration of the body.

It states that the sum of the forces acting on a body is equal to the mass of the body times its acceleration.

Centroidal Dynamics [5]

Orin et al. (2013) defined Centroidal dynamics as the dynamics of a humanoid robot projected at its CoM. ToDo

Example: Single Rigid Body Model [6]

The design of the Cheetah3 (MIT) has light limbs with low inertia as compared to the overall body, they simplify the control model to ignore the effects of the legs for planning ground reaction forces from the stance feet. They assumed the model as a Single Rigid Body Model (SRBM), and the controller model used as:

$$\underbrace{\begin{bmatrix} \mathbf{I}_3 & \cdots & \mathbf{I}_3 \\ [\boldsymbol{p}_1 - \boldsymbol{p}_c]\times & \cdots & [\boldsymbol{p}_4 - \boldsymbol{p}_c]\times \end{bmatrix}}_{\boldsymbol{A}} \boldsymbol{F} = \underbrace{\begin{bmatrix} m\left(\ddot{\boldsymbol{p}}_c + \boldsymbol{g}\right) \\ \boldsymbol{I}_G \dot{\boldsymbol{\omega}}_b \end{bmatrix}}_{\boldsymbol{b}} \tag{7}$$

This can be done by linearizing the Euler motion law equation, which offers advantages when using MPC formulation, as discussed by Kim et al. (2019) [7]. While the detailed formulation will not be covered, those interested in a deeper understanding are encouraged to refer to the cited paper.

Momentum Task

Momentum task is well described in [2, 8]. Koolen exploits the fact that the rate of change of whole-body centroidal momentum is both affine in the joint acceleration vector and linear in the external wrenches applied to the robot. This fact is used to formulate compact QP, solved at every control time step, which reconciles motion tasks in the form of constraints on the joint acceleration vector with the available contacts between the robot and its environment in the form of unilateral contacts or force-limited grasps.

$$\boldsymbol{h} = \begin{pmatrix} \boldsymbol{k} \\ \boldsymbol{l} \end{pmatrix} \in \mathbb{R}^6 \tag{8}$$

where, $\boldsymbol{h}$ denote centroidal momentum of the robot, $\boldsymbol{k} \in \mathbb{R}^3$ and $\boldsymbol{l} \in \mathbb{R}^3$ are the centroidal angular and linear momentum, respectively.

Since the centroidal dynamics of a robot are simple, no matter how complex the robot is, the rate of centroidal momentum is equal to the sum of all external wrenches applied to the robot,

$$\boldsymbol{h} = \boldsymbol{W}_g + \sum_i \boldsymbol{W}_{\mathrm{gr},i} + \sum_i \boldsymbol{W}_{\mathrm{ext},i} \tag{9}$$

where, $\boldsymbol{W}_g$ is the wrench due to gravity, and the $\boldsymbol{W}_{\mathrm{gr}}$ and $\boldsymbol{W}_{\mathrm{ext}}$ are ground reaction wrenches and external wrenches applied to the robot.

Orin and Goswami showed that there is a simple linear relationship between the joint velocity vector v and the robot's centroidal momentum[9]:

$$\boldsymbol{h} = \boldsymbol{A}(\boldsymbol{q})\dot{\boldsymbol{\xi}} \tag{10}$$

where, $\boldsymbol{A}(\boldsymbol{q})$ is called the centroidal momentum matrix (CMM). Substituting this into Eq. (9) results in

$$\boldsymbol{A}\ddot{\boldsymbol{\xi}} + \dot{\boldsymbol{A}}\dot{\boldsymbol{\xi}} = \boldsymbol{W}_g + \sum_i \boldsymbol{W}_{\mathrm{gr},i} + \sum_i \boldsymbol{W}_{\mathrm{ext},i} \tag{11}$$

which shows that there is an affine relationship between the joint acceleration vector $\dot{\boldsymbol{q}}$ and the external wrenches applied to the robot. Koolen exploits this fact in the proposed control framework.

Centroidal Momentum Matrix [9]

Orin et al. (2008) established the relationship between the Centroidal Momentum Matrix and the well-known joint-space inertia matrix.

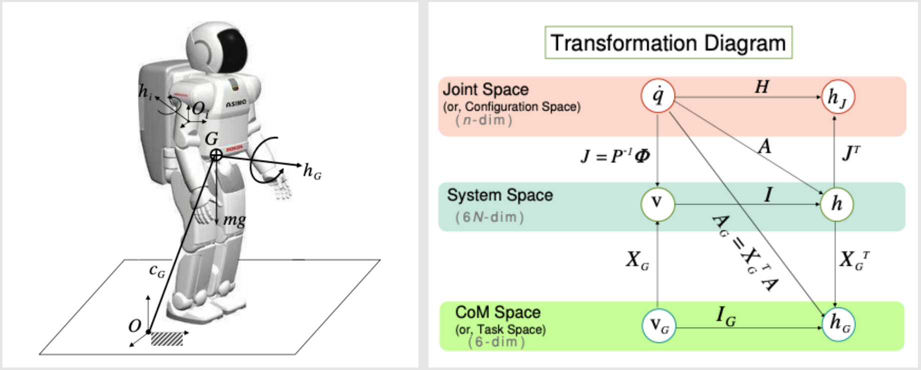

Figure 2: Humanoid showing link and centroidal momentum vectors. CMM Diagram of inter-relationships among the velocities and momenta of robot.

Motion Tracking Task

Motion tracking task can be presented as:

$$\boldsymbol{J}_{\mathrm{task}}\ddot{\boldsymbol{\xi}} + \dot{\boldsymbol{J}}_{\mathrm{task}}\dot{\boldsymbol{\xi}} = \ddot{\boldsymbol{x}}^d_{\mathrm{task}} \tag{12}$$

where, $\ddot{\boldsymbol{x}}$ is a desired task acceleration that will correspond to a desired closed-loop behavior, which are obtained from PD controller.

Example: Swing Foot Tracking

In the context of wbc, the following equation represents the relationship between joint acceleration and the desired swing foot acceleration:

$$\boldsymbol{J}_{\mathrm{sw}}\ddot{\boldsymbol{\xi}} + \dot{\boldsymbol{J}}_{\mathrm{sw}}\dot{\boldsymbol{\xi}} = \ddot{\boldsymbol{p}}^d_{\mathrm{sw}} \tag{13}$$

where, $\boldsymbol{J}_{\mathrm{sw}}$ denotes the swing foot Jacobian, $\ddot{\boldsymbol{\xi}}$ is the joint acceleration, $\dot{\boldsymbol{J}}_{\mathrm{sw}}$ represents the time derivative of the swing foot Jacobian, $\dot{\boldsymbol{q}}$ is the joint velocity, and $\ddot{\boldsymbol{p}}^d_{\mathrm{sw}}$ is the desired swing foot acceleration. This equation ensures that the joint accelerations are controlled to achieve the specified swing foot dynamics.

Regularization Task

The objective is to normalize and minimize joint acceleration and contact force, assigning relatively small weighting values to these parameters to ensure effective regularization in WBC.

Constraints

Floating-base Dynamics Constraints

To obey/ensure the floating-base dynamics are considered? (ToDo)

The floating-based dynamics equality constraints in whole-body control are essential for several reasons. Firstly, they ensure that the equations of motion for the entire robot, including its floating base, are accurately modeled and adhered to. This is crucial for maintaining the overall dynamic balance and stability of the robot, particularly when it interacts with the environment or performs complex maneuvers.

$$\boldsymbol{M}_b\ddot{\boldsymbol{\xi}} + \boldsymbol{C}_b\dot{\boldsymbol{\xi}} + \boldsymbol{g}_b = \boldsymbol{X}^T_{bC_k}\boldsymbol{\lambda}_{C_k} \tag{14}$$

or,

$$\boldsymbol{S}_f(\boldsymbol{M}\ddot{\boldsymbol{\xi}} + \boldsymbol{C}\dot{\boldsymbol{\xi}} + \boldsymbol{g}) = \boldsymbol{S}_f\boldsymbol{X}^T_{C_k}\boldsymbol{\lambda}_{C_k} \tag{15}$$

Contact Constraints

While the swing leg tracks the desired end-effector task; the remaining foot must remain in contact with the ground. Therefore, contact constraints are formulated as follows:

$$\boldsymbol{J}_c\ddot{\boldsymbol{\xi}} + \dot{\boldsymbol{J}}_c\dot{\boldsymbol{\xi}} = \boldsymbol{0} \tag{16}$$

where $\boldsymbol{J}_c$ is the contact jacobian and $\boldsymbol{\xi}$ represents the generalized coordinates. Reformulate Eq. (16) according to the decision variable:

$$\begin{bmatrix}\boldsymbol{J}_c & \boldsymbol{0}\end{bmatrix}\boldsymbol{z} = -\dot{\boldsymbol{J}}_c\dot{\boldsymbol{\xi}} \tag{17}$$

These are usually used as hard equality constraints that ensure the contact foot does not move and remains on the ground.

Joint Torque Limit Constraints

Since we use an optimization approach, the torque solution might be too large, which a real robot system cannot afford. Therefore, it is necessary to consider the joint torque limit constraints as follows for safety:

$$\boldsymbol{\tau}_{\min} \leq \boldsymbol{\tau} \leq \boldsymbol{\tau}_{\max} \tag{18}$$

We have to formulate the constraints above with the decision values; we can reformulate joint torque constraints as follows:

$$\boldsymbol{lb}_{\tau_{min}} \leq \boldsymbol{C}_\tau\boldsymbol{z} \leq \boldsymbol{ub}_{\tau_{max}} \tag{19}$$

where,

$$\begin{aligned}\boldsymbol{C}_\tau &= \begin{bmatrix}\boldsymbol{M}_j & -\boldsymbol{J}^T_{bC_k}\end{bmatrix}\\ \boldsymbol{ub}_{\tau_{max}} &= \boldsymbol{\tau}_{j_{\max}} - (\boldsymbol{C}_j\dot{\boldsymbol{\xi}} + \boldsymbol{g}_j)\\ \boldsymbol{lb}_{\tau_{min}} &= \boldsymbol{\tau}_{j_{\min}} - (\boldsymbol{C}_j\dot{\boldsymbol{\xi}} + \boldsymbol{g}_j)\end{aligned} \tag{20}$$

$\boldsymbol{S}_j$ is the joint selection matrix. $\boldsymbol{\tau}_{j_{\min}}$ and $\boldsymbol{\tau}_{j_{\max}}$ are the torque limit bounds that the user provides, respectively.

Friction Cone Constraints

To prevent the contact feet from slipping, coulomb friction theory and friction cone are used. Basically, we have to consider two conditions.

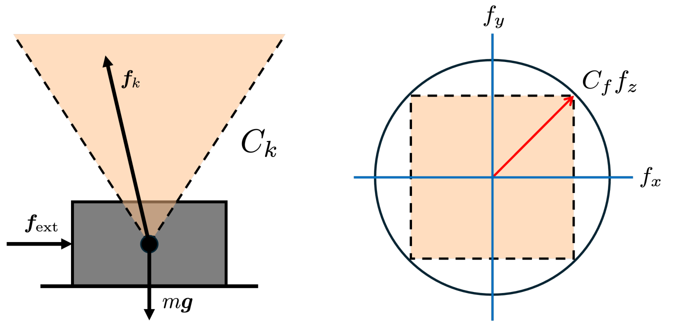

Figure 3: Coulomb friction cone and Linear Cone Approximation

Non-negative conditions: All the normal components of reactive force on the ground plane are not attractive but repulsive.

$$f_z \geq 0 \tag{21}$$

Friction Forces: The tangent component of reactive force on the ground plane is always within the friction cone.

$$f_x^2 + f_y^2 \leq C_f^2 f_z^2 \tag{22}$$

However, since the overall friction cone constraint conditions Eq. (23) are nonlinear,

$$\mathcal{F} = \left\{ f \in \Re^3 \mid f_z \geq 0, f_x^2 + f_y^2 \leq C_f^2 f_z^2 \right\} \tag{23}$$

we need to use the cone approximation technique [10] to approximate the friction cone as a friction pyramid or polyhedral(linear) friction cone.

1) Unilateral Point Contact with Coulomb Friction

$$f_z \geq 0, \quad |f_x| \leq \frac{C_f}{\sqrt{2}} f_z, \quad |f_y| \leq \frac{C_f}{\sqrt{2}} f_z \tag{24}$$

$$\begin{bmatrix} 1 & 0 & C_f/\sqrt{2} \\ -1 & 0 & C_f/\sqrt{2} \\ 0 & 1 & C_f/\sqrt{2} \\ 0 & -1 & C_f/\sqrt{2} \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} \geq 0 \tag{25}$$

2) Contact Forces Approximation in Contact Frame

Polyhedral Approximation

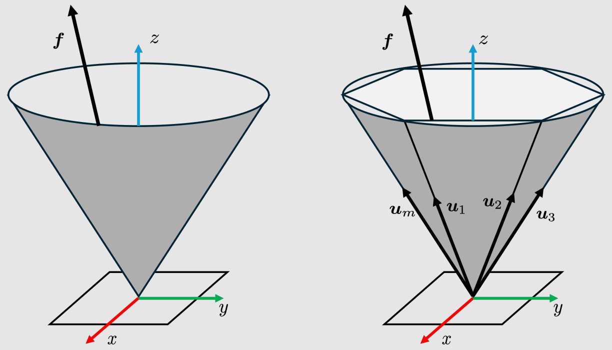

Figure 4: Friction Cone Approximation.

We could define an approximate quadratic cone as a polyhedral cone as the positive span of a number of edges of the approximate cone as shown in the figure.

$$\boldsymbol{f} = \boldsymbol{u}_1\alpha_1 + \boldsymbol{u}_2\alpha_2 + \cdots + \boldsymbol{u}_m\alpha_m = \boldsymbol{U}\boldsymbol{\alpha} \tag{26}$$

where,

$$\begin{aligned} \boldsymbol{U} &= \begin{bmatrix}\boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_m\end{bmatrix} \\ \boldsymbol{\alpha} &= \begin{bmatrix}\alpha_1 & \alpha_2 & \cdots & \alpha_m\end{bmatrix}^T \end{aligned} \tag{27}$$

m is the size of the polyhedral, and $\boldsymbol{u}_i$ is:

$$\boldsymbol{u}_i = \begin{bmatrix} \mu\cos\frac{2\pi(i-1/2)}{m} \\ \mu\sin\frac{2\pi(i-1/2)}{m} \\ 1 \end{bmatrix}. \tag{28}$$

$\boldsymbol{\alpha}$ is the intensity of the forces.

Using the polyhedral approximation method, the inequality constraints can be simply done by following criteria:

$$\boldsymbol{\alpha} \geq \mathbf{0} \tag{29}$$

and we can reformulate based on the decision variable $\boldsymbol{z} = (\ddot{\boldsymbol{\xi}}^T, \boldsymbol{\beta}^T)^T$ as:

$$\begin{bmatrix}\mathbf{0} & -\mathbf{I}\end{bmatrix}\boldsymbol{z} \leq \mathbf{0} \tag{30}$$

where, $\boldsymbol{\beta} = (\boldsymbol{\alpha}_0, \cdots, \boldsymbol{\alpha}_n)$, and n is the number of contact that robot currently have.

Example: Cone Approximation (Friction Pyramid)

When approximate friction cone as friction pyramid, m equals 4, then the forces can be defined as follows:

$$\boldsymbol{f} = \begin{bmatrix} \mu & 0 & -\mu & 0 \\ 0 & \mu & 0 & -\mu \\ 1 & 1 & 1 & 1 \end{bmatrix}\boldsymbol{\alpha} \tag{31}$$

ZMP Constrtiants

To ensure stationary contacts, the center of pressure (CoP) at each end-effector needs to reside in the interior of the end-effector's support polygon, as shown in Fig. 5.

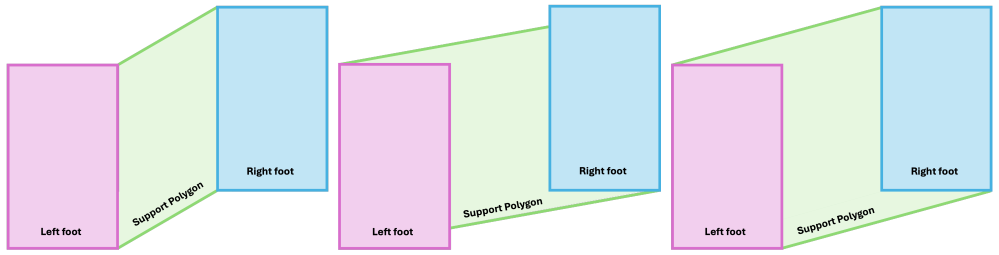

Figure 5: Inner Edge, Outer Edge, and Convex Hull

This can be expressed as a linear inequality by expressing the ground reaction force at the zero moment point.

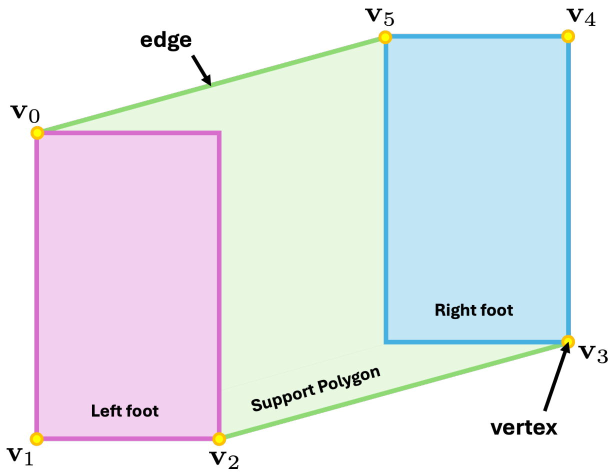

Figure 6: Convex Hull of Support Polygon

Firstly, we need to find the convex hull (Appendix 10.3) of the support polygon:

$$\boldsymbol{V} = \{\ \mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_n\ \} \tag{32}$$

where $\boldsymbol{V}$ is the set of vertices of the polygon. After finding the convex hull, set edge linear inequality constraints (a.k.a *halfspace* [11, 12]) using linear equation as Eq. (33)

$$ax + by + c \leq 0 \tag{33}$$

where,

$$\begin{aligned} a &= \mathbf{v}_{i,y} - \mathbf{v}_{j,y} \\ b &= \mathbf{v}_{i,x} - \mathbf{v}_{j,x} \\ c &= \mathbf{v}_{i,x}\mathbf{v}_{j,y} - \mathbf{v}_{j,x}\mathbf{v}_{i,y} \end{aligned} \tag{34}$$

where $\mathbf{v}_i$ and $\mathbf{v}_j$ are the current vertex and the next vertex. For checking if a Zero-Moment Point lies inside the support polygon, we can augment the edge equations and define inequality constraints as

$$\begin{bmatrix} a_0 & b_0 \\ \vdots & \vdots \\ a_n & b_n \end{bmatrix}\boldsymbol{p}_{\text{zmp}} \leq \begin{bmatrix} c_0 \\ \vdots \\ c_n \end{bmatrix} \tag{35}$$

Task Space Motion Limit Constraints?

It doesn't seen to be a necessary constraint(?). I believe it is needed when the self-collision problem has to be addressed.

Momentum-based Whole-body Control

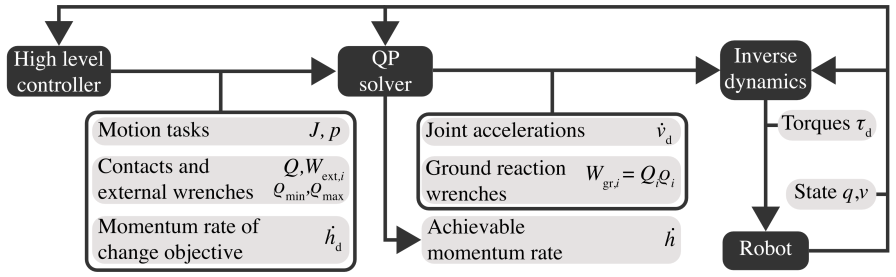
High level controller
QP solver
Inverse dynamics
Motion tasks J, p
Contacts and external wrenches $Q, W_{\text{ext},i}$ $\rho_{\min}, \rho_{\max}$
Momentum rate of change objective $\dot{h}_{\text{d}}$
Joint accelerations $\dot{v}_{\text{d}}$
Ground reaction wrenches $W_{\text{gr},i} = Q_i \rho_i$
Achievable momentum rate $\dot{h}$
Torques τ_{d}
State q,v
Robot

Figure 7: Proposed Controller framework at Koolen et al. (2020) IJHR.

Twan Koolen proposed a momentum-based whole-body controller [2, 8] using a weighted approach. The formulation is as follows:

$$\begin{aligned} \min_{\ddot{q}_d, \rho} \quad & (\boldsymbol{A}\ddot{\boldsymbol{q}} - \boldsymbol{b})^T \boldsymbol{C}_h (\boldsymbol{A}\ddot{\boldsymbol{q}} - \boldsymbol{b}) + \boldsymbol{\rho}^T \boldsymbol{C}_\rho \boldsymbol{\rho} + \ddot{\boldsymbol{q}}^T \boldsymbol{C}_{\ddot{q}} \ddot{\boldsymbol{q}} \\ \text{subject to} \quad & \boldsymbol{J}\ddot{\boldsymbol{q}} = \boldsymbol{p} \\ & \boldsymbol{A}\dot{\boldsymbol{v}}_d + \dot{\boldsymbol{A}}\boldsymbol{v} = \boldsymbol{W}_g + \boldsymbol{Q}\boldsymbol{\rho} + \sum \boldsymbol{W}_{\text{ext},i} \\ & \boldsymbol{\rho}_{\min} \leq \boldsymbol{\rho} \leq \boldsymbol{\rho}_{\max} \end{aligned} \tag{36}$$

where $\boldsymbol{b} = \dot{\boldsymbol{h}}_d - \dot{\boldsymbol{A}}\dot{\boldsymbol{q}}$, with $\dot{\boldsymbol{h}} \in \mathbb{R}^6$ a desired rate of change of centroidal momentum.

Task-Prioritized, *or* WBIC

The following approach [4, 7] calculates the nominal joint acceleration command using inverse kinematics with null space projection method [13]. Both approaches find the relaxation value that satisfies the equality and inequality constraints.

Kim et al. 2020 IJRR [4]

In [4] called this controller as Whole-body Locomotion Control (WBLC), and formulated as follows:

$$\begin{aligned} \min_{\mathbf{F}_r, \ddot{\mathbf{x}}_c, \delta_{\ddot{\mathbf{q}}}} \quad & \mathbf{F}_r^\top \boldsymbol{W}_r \mathbf{F}_r + \ddot{\mathbf{x}}_c^\top \boldsymbol{W}_c \ddot{\mathbf{x}}_c + \boldsymbol{\delta}_{\ddot{\mathbf{q}}}^\top \boldsymbol{W}_{\ddot{\mathbf{q}}} \delta_{\ddot{\mathbf{q}}} \\ \text{s.t.} \quad & \boldsymbol{U}\mathbf{F}_r \geq \mathbf{0} \\ & \boldsymbol{S}\mathbf{F}_r \leq \mathbf{F}_{r,z}^{\max} \\ & \ddot{\mathbf{x}}_c = \boldsymbol{J}_c \ddot{\mathbf{q}} + \dot{\boldsymbol{J}}_c \dot{\mathbf{q}}, \\ & \boldsymbol{A}\ddot{\mathbf{q}} + \mathbf{b} + \mathbf{g} = \begin{pmatrix} \mathbf{0}_{6\times1} \\ \boldsymbol{\tau}^{\text{cmd}} \end{pmatrix} + \boldsymbol{J}_c^\top \mathbf{F}_r \\ & \ddot{\mathbf{q}} = \ddot{\mathbf{q}}^{\text{cmd}} + \boldsymbol{\delta}_{\ddot{\mathbf{q}}} \\ & \ddot{\mathbf{q}}^{\text{cmd}} = \ddot{\mathbf{q}}^{\text{d}} + k_d(\dot{\mathbf{q}}^{\text{d}} - \dot{\mathbf{q}}) + k_p(\mathbf{q}^{\text{d}} - \mathbf{q}) \\ & \boldsymbol{\tau}_{\min} \leq \boldsymbol{\tau}^{\text{cmd}} \leq \boldsymbol{\tau}_{\max} \end{aligned} \tag{37}$$

Kim et al. 2019 arXiv [7]

In [7] called this controller as Whole-body Impulse Control (WBIC), and formulated as follows:

$$\begin{aligned} \min_{\boldsymbol{\delta}_{\boldsymbol{f}_r}, \boldsymbol{\delta}_{\boldsymbol{f}}} \quad & \boldsymbol{\delta}_{\boldsymbol{f}_r}^T \boldsymbol{Q}_1 \boldsymbol{\delta}_{\boldsymbol{f}_r} + \boldsymbol{\delta}_{\boldsymbol{f}}^T \boldsymbol{Q}_2 \boldsymbol{\delta}_{\boldsymbol{f}} \\ \text{s.t.} \quad & \boldsymbol{S}_f(\boldsymbol{A}\ddot{\mathbf{q}} + \mathbf{b} + \mathbf{g}) = \boldsymbol{S}_f \boldsymbol{J}_c^\top \mathbf{f}_r \\ & \ddot{\mathbf{q}} = \ddot{\mathbf{q}}^{\text{cmd}} + \begin{bmatrix} \boldsymbol{\delta}_f \\ \mathbf{0}_{n_j} \end{bmatrix} \\ & \mathbf{f}_r = \mathbf{f}_r^{\text{MPC}} + \boldsymbol{\delta}_{\mathbf{f}_r} \\ & \boldsymbol{W}\mathbf{f}_r \geq \mathbf{0}, \end{aligned} \tag{38}$$

Both approaches use a null-space projection skill and find relaxation values, but it has slight differences in the decision variables they are optimizing(minimizing) for. Also, I am still unsure which has better benefits for the humanoid robot locomotion control.

Hierarchical Optimization WBC

Hierarchical Optimization [14, 15, 3] has advantages such as ,

A general task T can be defined as of linear equality and/or inequality constraints on the solution vector $\boldsymbol{x} \in \mathbb{R}^n$:

$$\mathrm{T} : \begin{cases} \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} = \boldsymbol{v} \\ \boldsymbol{C}\boldsymbol{x} - \boldsymbol{d} \leq \boldsymbol{w} \end{cases} \tag{39}$$

and $\boldsymbol{v}$ and $\boldsymbol{w}$ are slack variables to be minimized, formulate it as a QP problem:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \frac{1}{2}\|\boldsymbol{v}\|^2 + \frac{1}{2}\|\boldsymbol{w}\|^2 \\ \text{subject to} \quad & \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} = \boldsymbol{v} \\ & \boldsymbol{C}\boldsymbol{x} - \boldsymbol{d} = \boldsymbol{w} \end{aligned} \tag{40}$$

also can be expressed as:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2 + \frac{1}{2}\|\boldsymbol{w}\|^2 \\ \text{subject to} \quad & \boldsymbol{C}\boldsymbol{x} - \boldsymbol{d} = \boldsymbol{w} \end{aligned} \tag{41}$$

it is equivalent QP problem with $\boldsymbol{X} = [\boldsymbol{b}^T \ \boldsymbol{w}^T]^T$ is:

$$\begin{aligned} \min_{\boldsymbol{X}} \quad & \frac{1}{2}\boldsymbol{X}^T \boldsymbol{Q} \boldsymbol{X} + \boldsymbol{p}^T \boldsymbol{X} \\ \text{subject to} \quad & \bar{\boldsymbol{C}}\boldsymbol{X} \leq \bar{\boldsymbol{d}} \end{aligned} \tag{42}$$

where $\boldsymbol{Q}$ and $\boldsymbol{p}$ can be calculated from $\boldsymbol{Q} = \bar{\boldsymbol{A}}^T \bar{\boldsymbol{A}}$ and $\boldsymbol{p} = -\bar{\boldsymbol{A}}^T \bar{\boldsymbol{b}}$. The expressions of each term are defined as:

$$\begin{aligned} & \bar{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{A} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix}, \ \bar{\boldsymbol{b}} = \begin{bmatrix} -\boldsymbol{b} \\ \mathbf{0}, \end{bmatrix} \\ & \bar{\boldsymbol{C}} = [\boldsymbol{C} \ \ -\boldsymbol{I}], \ \bar{\boldsymbol{d}} = \boldsymbol{d} \end{aligned} \tag{43}$$

Other Methods

Feature-based Locomotion Controllers [16]

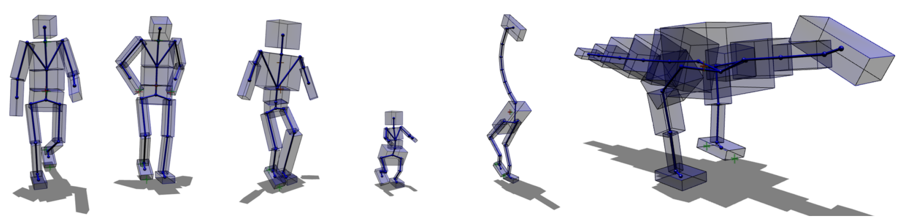

Figure 8: Figure depicted at the feature-based locomotion controllers.

According to the paper, high-level features include task-relevant quantities such as CoM, Angular Momentum, and End-Effector motion.

1) Objectives

Each objective is formulated in terms of some feature $\boldsymbol{y}$ of the motion.
Setpoint Objective: This objective applies linear control to some feature of pose.

$$\boldsymbol{y} = f(\boldsymbol{q}) \tag{44}$$

Target Objective:
Angular Momentum (AM) Objective: Regulating AM improves balance stability and robustness. Beyond the limitation of requiring reference COP, they proposed an alternate AM control formulation that produces not only balance but also jumping and walking.
Minimum Torque Objective

2) Prioritized Optimization

Dynamic Balance Force Control [17]

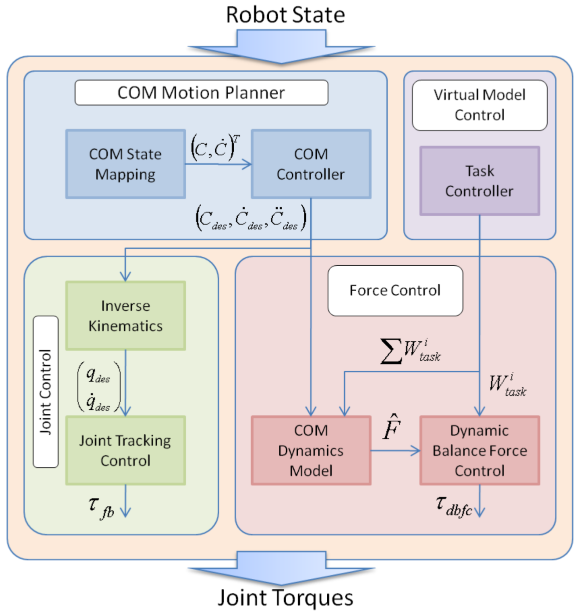

Figure 9: Proposed DBFC Controller.

This paper presents a model-based method called Dynamic Balance Force Control (DBFC) for determining full-body joint torques based on desired COM motion and contact forces for compliant humanoid robots.

Passivity-based Whole-body Control [18]

Simple Weighted Approach [19]

In [19], it also uses weighted approaches to complete the motion track with several given constraints.

Cafe-MPC, Value-Function based WBC [20]

Apendix

Basic Quadratic Programming formulation

A general quadratic programming (QP) problem has the form:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{p}^T\boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{C}\boldsymbol{x} \leq \boldsymbol{d} \end{aligned} \tag{45}$$

If you have one task with the cost term of a least square form:

$$\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2 \tag{46}$$

It can be transformed into a QP problem:

$$\begin{aligned} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2 &= \frac{1}{2}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})^T(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) \\ &= \frac{1}{2}(\boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{b} - \boldsymbol{b}^T\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{b}) \\ &= \frac{1}{2}(\boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} - 2\boldsymbol{b}^T\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{b}) \end{aligned} \tag{47}$$

since $\boldsymbol{b}^T\boldsymbol{b}$ is a fixed term, it can be dropped out; therefore, we could result QP problem having cost terms:

$$\begin{aligned} \boldsymbol{Q} &= \boldsymbol{A}^T\boldsymbol{A} \\ \boldsymbol{p} &= (-\boldsymbol{b}^T\boldsymbol{A})^T = -\boldsymbol{A}^T\boldsymbol{b} \end{aligned} \tag{48}$$

If there are multiple tasks, the cost term of least square has the form:

$$\frac{1}{2}\|\boldsymbol{A}_1\boldsymbol{x} - \boldsymbol{b}_1\|^2 + \frac{1}{2}\|\boldsymbol{A}_2\boldsymbol{x} - \boldsymbol{b}_2\|^2 \tag{49}$$

expanding it:

$$\begin{aligned} &= \frac{1}{2}(\boldsymbol{A}_1\boldsymbol{x} - \boldsymbol{b}_1)^T(\boldsymbol{A}_1\boldsymbol{x} - \boldsymbol{b}_1) + \frac{1}{2}(\boldsymbol{A}_2\boldsymbol{x} - \boldsymbol{b}_2)^T(\boldsymbol{A}_2\boldsymbol{x} - \boldsymbol{b}_2) \\ &= \frac{1}{2}(\boldsymbol{x}^T\boldsymbol{A}_1^T\boldsymbol{A}_1\boldsymbol{x} - 2\boldsymbol{b}_1^T\boldsymbol{A}_1\boldsymbol{x} + \boldsymbol{b}_1^T\boldsymbol{b}_1) \\ &\qquad + \frac{1}{2}(\boldsymbol{x}^T\boldsymbol{A}_2^T\boldsymbol{A}_2\boldsymbol{x} - 2\boldsymbol{b}_2^T\boldsymbol{A}_2\boldsymbol{x} + \boldsymbol{b}_2^T\boldsymbol{b}_2) \\ &= \frac{1}{2}\left[\boldsymbol{x}^T(\boldsymbol{A}_1^T\boldsymbol{A}_1 + \boldsymbol{A}_2^T\boldsymbol{A}_2)\boldsymbol{x} - 2(\boldsymbol{b}_1^T\boldsymbol{A}_1 + \boldsymbol{b}_2^T\boldsymbol{A}_2)\boldsymbol{x} + \boldsymbol{b}_1^T\boldsymbol{b}_1 + \boldsymbol{b}_2^T\boldsymbol{b}_2\right] \end{aligned} \tag{50}$$

therefore, we could result in QP cost term as:

$$\begin{aligned} \boldsymbol{Q} &= \boldsymbol{A}^T\boldsymbol{A} = [\boldsymbol{A}_1^T \ \boldsymbol{A}_2^T]\begin{bmatrix}\boldsymbol{A}_1 \\ \boldsymbol{A}_2\end{bmatrix} = \sum_i \boldsymbol{A}_i^T\boldsymbol{A}_i \\ \boldsymbol{p} &= (-\boldsymbol{b}^T\boldsymbol{A})^T = -\boldsymbol{A}^T\boldsymbol{b} = -[\boldsymbol{A}_1^T \ \boldsymbol{A}_2^T]\begin{bmatrix}\boldsymbol{b}_1 \\ \boldsymbol{b}_2\end{bmatrix} = -\sum_i \boldsymbol{A}_i^T\boldsymbol{b}_i \end{aligned} \tag{51}$$

The general problem cost term with n weighted tasks:

$$\frac{1}{2}\omega_1\|\boldsymbol{A}_1\boldsymbol{x} - \boldsymbol{b}_1\|^2 + \frac{1}{2}\omega_2\|\boldsymbol{A}_2\boldsymbol{x} - \boldsymbol{b}_2\|^2 + \cdots \frac{1}{2}\omega_n\|\boldsymbol{A}_n\boldsymbol{x} - \boldsymbol{b}_n\|^2 \tag{52}$$

we could result in general QP cost term as:

$$\begin{aligned} \boldsymbol{A} &= \begin{bmatrix}\omega_1\boldsymbol{A}_1 \\ \omega_2\boldsymbol{A}_2 \\ \vdots \\ \omega_n\boldsymbol{A}_n\end{bmatrix}, \ \boldsymbol{b} = \begin{bmatrix}\omega_1\boldsymbol{b}_1 \\ \omega_2\boldsymbol{b}_2 \\ \vdots \\ \omega_n\boldsymbol{b}_n\end{bmatrix} \\ \boldsymbol{Q} &= \boldsymbol{A}^T\boldsymbol{A} = \sum_i \omega_i\boldsymbol{A}_i^T\boldsymbol{A}_i \\ \boldsymbol{p} &= -\boldsymbol{A}^T\boldsymbol{b} = -\sum_i \omega_i\boldsymbol{A}_i^T\boldsymbol{b}_i \end{aligned} \tag{53}$$

If there are two tasks with two different variables. The overall cost term is defined as:

$$\frac{1}{2}\|\boldsymbol{A}_1\boldsymbol{x}_1 - \boldsymbol{b}_1\|^2 + \frac{1}{2}\|\boldsymbol{A}_2\boldsymbol{x}_2 - \boldsymbol{b}_2\|^2 \tag{54}$$

expanding it:

$$\begin{aligned} &= \frac{1}{2}(\boldsymbol{A}_1\boldsymbol{x}_1 - \boldsymbol{b}_1)^T(\boldsymbol{A}_1\boldsymbol{x}_1 - \boldsymbol{b}_1) + (\boldsymbol{A}_2\boldsymbol{x}_2 - \boldsymbol{b}_2)^T(\boldsymbol{A}_2\boldsymbol{x}_2 - \boldsymbol{b}_2) \\ &= \frac{1}{2}(\boldsymbol{x}_1\boldsymbol{A}_1^T\boldsymbol{A}\boldsymbol{x}_1 - 2\boldsymbol{b}_1^T\boldsymbol{A}_1\boldsymbol{x}_1 + \boldsymbol{b}_1^T\boldsymbol{b}_1) \\ &\qquad + \frac{1}{2}(\boldsymbol{x}_2\boldsymbol{A}_2^T\boldsymbol{A}\boldsymbol{x}_2 - 2\boldsymbol{b}_2^T\boldsymbol{A}_2\boldsymbol{x}_2 + \boldsymbol{b}_2^T\boldsymbol{b}_2) \\ &= \frac{1}{2}\Bigg([\boldsymbol{x}_1 \ \boldsymbol{x}_2]\begin{bmatrix}\boldsymbol{A}_1^T\boldsymbol{A}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_2^T\boldsymbol{A}_2\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_1 \\ \boldsymbol{x}_2\end{bmatrix} \\ &\qquad -2\left[\boldsymbol{b}_1^T\boldsymbol{A}_1 \quad \boldsymbol{b}_2^T\boldsymbol{A}_2\right]\begin{bmatrix}\boldsymbol{x}_1 \\ \boldsymbol{x}_2\end{bmatrix} + \boldsymbol{b}_1^T\boldsymbol{b}_1 + \boldsymbol{b}_2^T\boldsymbol{b}_2\Bigg) \\ &= \frac{1}{2}\Bigg([\boldsymbol{x}_1 \ \boldsymbol{x}_2]\begin{bmatrix}\boldsymbol{A}_1^T & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_2^T\end{bmatrix}\begin{bmatrix}\boldsymbol{A}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_2\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_1 \\ \boldsymbol{x}_2\end{bmatrix} \\ &\qquad -2[\boldsymbol{b}_1^T \ \boldsymbol{b}_2^T]\begin{bmatrix}\boldsymbol{A}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_2\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_1 \\ \boldsymbol{x}_2\end{bmatrix} + [\boldsymbol{b}_1^T \ \boldsymbol{b}_2^T]\begin{bmatrix}\boldsymbol{b}_1 \\ \boldsymbol{b}_2\end{bmatrix}\Bigg) \end{aligned} \tag{55}$$

therefore, we could result in QP cost term as:

$$\begin{aligned} \boldsymbol{Q} &= \boldsymbol{A}^T\boldsymbol{A} = \begin{bmatrix}\boldsymbol{A}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_2\end{bmatrix}^T\begin{bmatrix}\boldsymbol{A}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_2\end{bmatrix} \\ \boldsymbol{p} &= (-\boldsymbol{b}^T\boldsymbol{A})^T = -\boldsymbol{A}^T\boldsymbol{b} = -\begin{bmatrix}\boldsymbol{A}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_2\end{bmatrix}^T\begin{bmatrix}\boldsymbol{b}_1 \\ \boldsymbol{b}_2\end{bmatrix} \end{aligned} \tag{56}$$

The general cost term with n weighted tasks:

$$\frac{1}{2}\omega_1\|\boldsymbol{A}_1\boldsymbol{x}_1 - \boldsymbol{b}_1\|^2 + \frac{1}{2}\omega_2\|\boldsymbol{A}_2\boldsymbol{x}_2 - \boldsymbol{b}_2\|^2 + \cdots \frac{1}{2}\omega_n\|\boldsymbol{A}_n\boldsymbol{x}_n - \boldsymbol{b}_n\|^2 \tag{57}$$

we could result in general QP cost term as:

$$\begin{aligned} \boldsymbol{A} &= \begin{bmatrix}\omega_1\boldsymbol{A}_1 & & & \\ & \omega_2\boldsymbol{A}_2 & & \\ & & \ddots & \\ & & & \omega_n\boldsymbol{A}_n\end{bmatrix}, \ \boldsymbol{b} = \begin{bmatrix}\omega_1\boldsymbol{b}_1 \\ \omega_2\boldsymbol{b}_2 \\ \vdots \\ \omega_n\boldsymbol{b}_n\end{bmatrix} \\ \boldsymbol{x} &= \begin{bmatrix}\boldsymbol{x}_1^T & \boldsymbol{x}_2^T & \cdots & \boldsymbol{x}_n^T\end{bmatrix}^T \end{aligned} \tag{58}$$

Screw Transformation

Convex Hull

1) Graham Scan algorithm

https://david0506.tistory.com/62

2) Andrew's monotone chain algorithm

Implements Andrew's monotone chain algorithm to find the convex hull of a set of points:

1. Sort Points: The points are sorted lexicographically (first by x-coordinate, then by y-coordinate if x is the same).

2. Lower Hull: Iterate through the sorted points to build the lower hull. For each point, if adding it to the hull would cause a clockwise turn, remove the last point from the hull.

3. Upper Hull: Iterate through the points in reverse order to build the upper hull similarly.

4. Combine Hulls: The lower and upper hulls are combined to form the convex hull.

The result is a vector of points forming the vertices of the convex hull.

useful solvers

- QP (quadprog, qpOASES, OSQP, Gurobi, ProxQP)
- General Nonlinear Programming solvers (IPOPT, SNOPT)

References

[1] Federico L Moro and Luis Sentis. Whole-body control of humanoid robots. *Humanoid robotics: a reference*, pages 1161–1183, 2019.

[2] Twan Koolen, Sylvain Bertrand, Gray Thomas, Tomas De Boer, Tingfan Wu, Jesper Smith, Johannes Englsberger, and Jerry Pratt. Design of a momentum-based control framework and application to the humanoid robot atlas. *International Journal of Humanoid Robotics*, 13(01):1650007, 2016.

[3] Alexander Herzog, Nicholas Rotella, Sean Mason, Felix Grimminger, Stefan Schaal, and Ludovic Righetti. Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid. *Autonomous Robots*, 40:473–491, 2016.

[4] Donghyun Kim, Steven Jens Jorgensen, Jaemin Lee, Junhyeok Ahn, Jianwen Luo, and Luis Sentis. Dynamic locomotion for passive-ankle biped robots and humanoids using whole-body locomotion control. *The International Journal of Robotics Research*, 39(8):936–956, 2020.

[5] David E Orin, Ambarish Goswami, and Sung-Hee Lee. Centroidal dynamics of a humanoid robot. *Autonomous robots*, 35:161–176, 2013.

[6] Gerardo Bledt, Matthew J Powell, Benjamin Katz, Jared Di Carlo, Patrick M Wensing, and Sangbae Kim. Mit cheetah 3: Design and control of a robust, dynamic quadruped robot. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2245–2252. IEEE, 2018.

[7] Donghyun Kim, Jared Di Carlo, Benjamin Katz, Gerardo Bledt, and Sangbae Kim. Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control. *arXiv preprint arXiv:1909.06586*, 2019.

[8] Frans Anton Koolen. *Balance control and locomotion planning for humanoid robots using nonlinear centroidal models*. PhD thesis, Massachusetts Institute of Technology, 2020.

[9] David E Orin and Ambarish Goswami. Centroidal momentum matrix of a humanoid robot: Structure and properties. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 653–659. IEEE, 2008.

[10] Milutin Nikolić, Branislav Borovac, and Mirko Raković. Dynamic balance preservation and prevention of sliding for humanoid robots in the presence of multiple spatial contacts. *Multibody System Dynamics*, 42:197–218, 2018.

[11] ConexOptimization Summary in Korean. Lie theory. https://convex-optimization-for-all.github.io/contents/chapter02/2021/02/11/02_02_01_Convex_sets_examples/.

[12] Scaron. Polyheda and polyopes. https://scaron.info/blog/polyhedra-and-polytopes.html.

[13] Donghyun Kim, Jaemin Lee, Junhyeok Ahn, Orion Campbell, Hochul Hwang, and Luis Sentis. Computationally-robust and efficient prioritized whole-body controller with contact constraints. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2018.

[14] C Dario Bellicoso, Christian Gehring, Jemin Hwangbo, Péter Fankhauser, and Marco Hutter. Perception-less terrain adaptation through whole body control and hierarchical optimization. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 558–564. IEEE, 2016.

[15] SongYan Xin. Hierarchical quadratic programming. https://sites.google.com/site/xinsongyan/research/hierarchical-qp.

[16] Martin De Lasa, Igor Mordatch, and Aaron Hertzmann. Feature-based locomotion controllers. *ACM transactions on graphics (TOG)*, 29(4):1–10, 2010.

[17] Benjamin J Stephens and Christopher G Atkeson. Dynamic balance force control for compliant humanoid robots. In *2010 IEEE/RSJ international conference on intelligent robots and systems*, pages 1248–1255. IEEE, 2010.

[18] Bernd Henze, Maximo A Roa, and Christian Ott. Passivity-based whole-body balancing for torque-controlled humanoid robots in multi-contact scenarios. *The International Journal of Robotics Research*, 35(12):1522–1543, 2016.

[19] Wenhan Cai, Qingkai Li, Songrui Huang, Hongjin Zhu, Yong Yang, and Mingguo Zhao. Squat motion of a bipedal robot using real-time kinematic prediction and whole-body control. *IET Cyber-Systems and Robotics*, 4(4):298–312, 2022.

[20] He Li and Patrick M Wensing. Cafe-mpc: A cascaded-fidelity model predictive control framework with tuning-free whole-body control. *arXiv preprint arXiv:2403.03995*, 2024.